

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-031155

(43)Date of publication of application : 02.02.1999

(51)Int.Cl.

G06F 17/30  
G06F 3/153  
G06F 12/00  
G06F 13/00

(21)Application number : 09-188333

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 14.07.1997

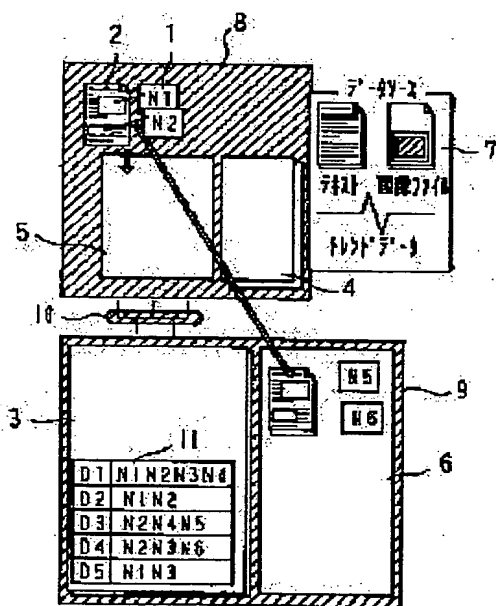
(72)Inventor : NAKAGAWA KOICHI

## (54) DATA PRESENTATION SYSTEM THROUGH WEB BROWSER

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To minimize data traffic in a system which presents data that is managed by a web server through a web browser that is connected by an intranet.

**SOLUTION:** A stub program 4 that manages data source 7 en bloc is provided on a server machine 8, the value of the source 7 is read from the program 4 onto a client machine 9, and a proxy program that retrieves a program part which needs the value and sends it is provided. Therefore, a program part on the machine 9 does not separately access the machine 8 and can read data from local memory of the proxy program, and data communication traffic through a web browser 6 can be reduced.



## LEGAL STATUS

[Date of request for examination] 25.09.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3606007

[Date of registration] 15.10.2004

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

**BEST AVAILABLE COPY**

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-31155

(43) 公開日 平成11年(1999) 2月2日

(51) Int.Cl.<sup>6</sup>

識別記号

F I

G 0 6 F 17/30

3/153

12/00

13/00

3 3 0

5 4 5

3 5 1

G 0 6 F 15/40

3/153

12/00

13/00

15/40

3 1 0 D

3 3 0 A

5 4 5 M

3 5 1 H

3 1 0 F

審査請求 未請求 請求項の数 3 O L (全 9 頁)

(21) 出願番号

特願平9-188333

(22) 出願日

平成9年(1997) 7月14日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 中川 晃一

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

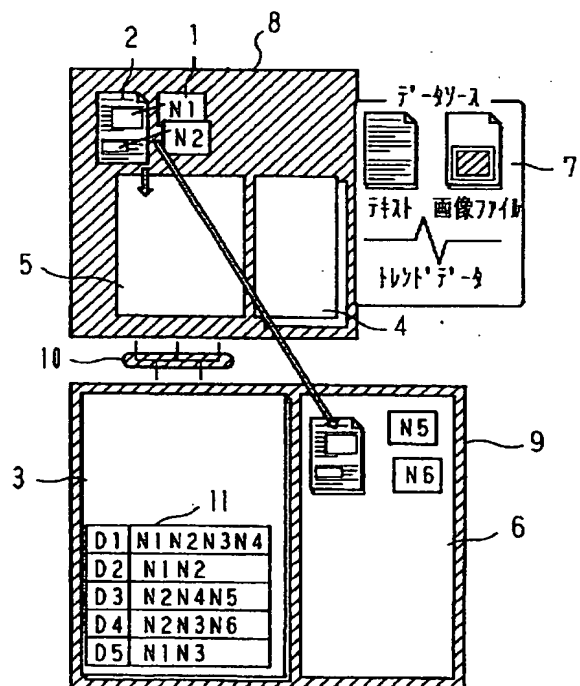
(74) 代理人 弁理士 宮田 金雄 (外2名)

(54) 【発明の名称】 ウェブ・ブラウザを介したデータ提示システム

(57) 【要約】

【課題】 ウェブ・サーバが管理するデータを、インターネットで結合されたウェブ・ブラウザを介して提示するシステムにおいて、データ通信量を最小限にする。

【解決手段】 サーバマシン上に、データソースを一括して管理するスタブプログラムを設け、クライアントマシン上に、スタブプログラムからデータソースの値を読み込み、その値を必要としているプログラム部分を検索して送信できるプロキシプログラムを設けた。そのためクライアントマシン上のプログラム部品は個別にサーバマシンにアクセスすることなく、プロキシプログラムのローカルメモリからデータを読みとることができ、ウェブ・ブラウザを介してのデータ通信量が削減できる。



## 【 特許請求の範囲】

【 請求項1 】 ウェブ・サーバが管理するデータをイントラネット で結合されたウェブ・ブラウザを介して提示するデータ提示システムにおいて、サーバマシン上に設けられ、ネットワーク上のデータソースの値の読み込みと更新を管理し、クライアントマシン上のプログラムへのデータの送信と更新の通知を行うスタブプログラムと、クライアントマシン上に設けられ、ウェブ・サーバから受信したドキュメント中に含まれる接続可能NLO部品のデータソースへのアクセスを管理するプロキシNLO部品と、前記プロキシNLO部品上に作成され、指定されたデータソースの識別子と、該データソースを使用するすべての接続可能NLO部品の識別子を含む接続可能NLO部品検索テーブルとを備え、前記プロキシNLO部品は、前記スタブプログラムへ前記接続可能NLO部品検索テーブル上のすべてのデータソースの識別子を前記スタブプログラムに送信し、前記スタブプログラムは前記プロキシNLO部品から受信したデータソース識別子に対応するデータが更新されたとき、当該データについて前記プロキシNLO部品に対してデータ値の送信とデータ値の更新の通知を行うことを特徴とするウェブ・ブラウザを介したデータ提示システム。

【 請求項2 】 前記接続可能NLO部品検索テーブルに、接続可能NLO部品が必要とするデータ更新周期と、当該接続可能NLO部品の最新のデータ更新時刻とを格納し、システムの現在時刻が前記最新のデータ更新時刻から前記データ更新周期だけ経過している時にのみ、前記プロキシNLO部品から前記接続可能NLO部品への更新通知とデータの読み込みを行うことを特徴とする請求項1記載のウェブ・ブラウザを介したデータ提示システム。

【 請求項3 】 前記接続可能NLO部品は、必要とするデータソースを前記プロキシNLO部品に指定した時に、同一のデータソースを直前に使用することを指定した他の接続可能NLO部品の識別子を受け取り、前記プロキシNLO部品は、同一のデータソースを最後に使用する接続可能NLO部品の識別子を前記接続可能NLO部品検索テーブルに格納し、データソースが更新された時にプロキシNLO部品は前記同一のデータソースを最後に使用する指定を行った接続可能NLO部品にのみ、更新の通知と更新データ値の読み込みを行い、当該接続可能NLO部品は、前記同一のデータソースを使用する他の接続可能NLO部品へ更新データ値を伝播させることを特徴とする請求項1記載のウェブ・ブラウザを介したデータ提示システム。

## 【 発明の詳細な説明】

## 【 0001 】

【 発明の属する技術分野】 本発明は、イントラネットに接続したコンピュータ間でウェブ・サーバとウェブ・ブ

ラウザを使ってイントラネット上にあるデータソースを提示するシステムに関するものである。

## 【 0002 】

【 従来の技術】 図12は従来のウェブブラウザを使ったデータ提示システムの例を示したものである。以下に図中の要素に関する説明をする。

・ NLO部品( 図中121 ) Network Loadable Objectの略。ウェブ・サーバからクライアントマシンへダウンロードできるプログラム。テキストや画像やトレンドデータなどを扱うことができる。例としてはサンマイクロシステム(株)のJava アプレットやマイクロソフト(株)のActiveXコントロールなどがある。

・ HTMLドキュメント( 図中122 ) HTML言語で記述されたテキストデータ。データの中にNLO部品を識別する情報も含むことができる。

・ ウェブ・サーバ( 図中123 ) サーバマシンで実行され、HTMLドキュメントやNLO部品をダウンロードするプログラム。

・ ウェブ・ブラウザ( 図中124 ) クライアントマシンで実行され、HTMLドキュメントを表示し、表示に必要なNLO部品をウェブ・サーバからダウンロードできるプログラム。

・ データソース( 図中125 ) テキストや画像などの様々な形式の値。動的に変化することもある。

・ サーバマシン( 図中126 ) ウェブ・サーバを実行し、NLO部品やHTMLドキュメントを保持する機能をもつコンピュータ。

・ クライアントマシン( 図中127 ) ウェブ・ブラウザを実行するコンピュータ。

・ イントラネット( 図中128 ) 社内や所内など閉じたコンピュータネットワーク。サーバマシンとクライアントマシンはこのネットワークに繋がれている。

【 0003 】 従来のウェブ・ブラウザを介したデータ提示システムの動作について図13に示すフローチャートに従って説明する。

( ステップ1301 ) まずユーザがウェブ・ブラウザに表示したいHTMLドキュメントを含むURLをウェブ・ブラウザに指定する。

( ステップ1302 ) ウェブ・サーバは指定されたURLから接続するウェブ・サーバのアドレスを取り出す。

( ステップ1303 ) ウェブ・サーバはアドレスに指定されたウェブサーバに接続する。

( ステップ1304 ) ウェブ・サーバはURLに指定されたHTMLドキュメントをウェブ・ブラウザに送信する。

( ステップ1305 ) ウェブ・ブラウザは送信されたHTMLドキュメントを受信する。

( ステップ1306 ) ウェブ・ブラウザは受信したHTMLドキュメントの中にNLO部品が含まれているか

を検査する。

(ステップ1307) ステップ1406においてNL O部品が含まれていれば、ウェブ・サーバからNL O部品をダウンロードする。

(ステップ1308) NL O部品は自分自身を初期化してブラウザと同じプロセス内で実行される。

【0004】次に、こうしたNL O部品が一定周期で更新されるプロセスデータをデータソースとして、その値を読み込む場合について図14に示すフローチャートを使って説明する。ここでプロセスデータとは発電所や工場などの産業プラントにおける監視用のセンサの測定値や、アクチュエータへの制御値などが代表的な例である。

$$t = T_c - T_o$$

【0006】(ステップ1405) 経過した時間 $t$ がデータソースの更新周期 $T_{Rate}$ より長い時間経過しているかどうかを検査する。

(ステップ1406) ステップ1405において、経過した時間がデータソースの更新時間より長ければ前回更新時刻 $T_o$ を現在時刻の値に更新する。

(ステップ1407) NL O部品がデータソースから値を読み込む。

(ステップ1408) ステップ1405において経過した時間がデータソースの更新時間に満たない場合は待機時間 $T_w$ を算出する。この時間は次にデータソースを読み込むまでの時間となるので次ぎの式で算出することができる。

$$T_w = t - T_{Rate} \quad \text{式(2)}$$

(ステップ1409) 時間 $T_w$ だけデータソース読込を待機してステップ1403に戻る。

【0007】

【発明が解決しようとする課題】従来の方法では1ページに使われるデータソースの種類に関わらず、使われるデータソースの数が多くなるにつれてNL O部品が必要とする通信の量が多くなる。一般的にはHTMLドキュメントの同じページ内に同じデータソースを要求する複数のNL O部品があることが多い、例えば同じデータソースの値をトレンドグラフのNL O部品で表示したり、データソース値をもとに計算した値を表示するメタがあったりする。このとき2つのNL O部品は同じデータソースに対して別々に値を読みにいかなければならなかった。

【0008】

【課題を解決するための手段】ウェブ・サーバが管理するデータをイントラネット で結合されたウェブ・ブラウザを介して提示するデータ提示システムにおいて、サーバマシン上に設けられ、ネットワーク上のデータソースの値の読み込みと更新を管理し、クライアントマシン上のプログラムへのデータの送信と更新の通知を行うスタブプログラムと、クライアントマシン上に設けられ、ウ

\*【0005】(ステップ1401) NL O部品が自分自身を初期化する。

(ステップ1402) 前回更新時刻 $T_o$ を設定する。前回更新時刻とはNL O部品が最後にデータソースの値を読み込んだ時刻である。この時刻にはデータの読込完了時刻と、読込開始時刻の2つがあるが、ここでは説明のしやすさから読込み開始時刻をこの時刻としている。この時刻の初期値はシステムの現在時刻である。

(ステップ1403) NL O部品がシステムから現在時刻を $T_c$ に代入する。

(ステップ1404) 前回の更新時刻から経過した時間を $t$ に代入する。経過した時間 $t$ は次の式によって算出できる。

$$\text{式(1)}$$

ウェブ・サーバから受信したドキュメント中に含まれる接続可能NL O部品のデータソースへのアクセスを管理するプロキシNL O部品と、前記プロキシNL O部品上に作成され、指定されたデータソースの識別子と、該データソースを使用するすべての接続可能NL O部品の識別子を含む接続可能NL O部品検索テーブルとを備え、前記プロキシNL O部品は、前記スタブプログラムへ前記接続可能NL O部品検索テーブル上のすべてのデータソースの識別子を前記スタブプログラムに送信し、前記スタブプログラムは前記プロキシNL O部品から受信したデータソース識別子に対応するデータが更新されたとき、当該データについて前記プロキシNL O部品に対してデータ値の送信とデータ値の更新の通知を行うことを特徴とするウェブ・ブラウザを介したデータ提示システムを設けたこと。

【0009】また、前記接続可能NL O部品検索テーブルに、接続可能NL O部品が必要とするデータ更新周期と、当該接続可能NL O部品の最新のデータ更新時刻とを格納し、システムの現在時刻が前記最新のデータ更新時刻から前記データ更新周期だけ経過している時のみ、前記プロキシNL O部品から前記接続可能NL O部品への更新通知とデータの読み込みを行うことを特徴とする請求項1記載のウェブ・ブラウザを介したデータ提示システム設けたこと。

【0010】また、前記接続可能NL O部品は、必要とするデータソースを前記プロキシNL O部品に指定した時に、同一のデータソースを直前に使用することを指定した他の接続可能NL O部品の識別子を受け取り、前記プロキシNL O部品は、同一のデータソースを最後に使用する接続可能NL O部品の識別子を前記接続可能NL O部品検索テーブルに格納し、データソースが更新された時にプロキシNL O部品は前記同一のデータソースを最後に使用する指定を行った接続可能NL O部品にのみ、更新の通知と更新データ値の読み込みを行い、当該接続可能NL O部品は、前記同一のデータソースを使用する他の接続可能NL O部品へ更新データ値を伝播

させることを特徴とする請求項1記載のウェブ・ブラウザを介したデータ提示システムを設けたこと。

【0011】

【発明の実施の形態】

実施の形態1. 図1は本発明の実施例である。以下に図中の構成要素について説明する。

・接続可能NLO部品(図中1) ウェブ・サーバからクライアントマシンへダウンロードできるプログラム。NLO部品はお互いに接続可能である。

・HTMLドキュメント(図中2) HTML言語で記述されたテキストデータ。データの中にNLO部品を識別する情報も含むことができる。

・プロキシ-NLO部品(図中3) NLO部品と接続でき、NLO部品が使用するデータソースに関する識別する情報を記録することができるNLO部品。

・スタブプログラム(図中4) サーバマシン上で実行され、ネットワーク上のデータソースの値を読み込んだり、値を更新することができるプログラム。プロキシ-NLO部品と接続して、プロキシ-NLO部品が登録したデータソースが更新されると、プロキシ-NLO部品に通知することもできる。

・ウェブ・サーバ(図中5) サーバマシンで実行され、HTMLドキュメントやNLO部品をダウンロードするプログラム。

・ウェブ・ブラウザ(図中6) クライアントマシン上で実行され、HTMLドキュメントを表示し、表示に必要なNLO部品をウェブ・サーバからダウンロードできるプログラム。

・データソース(図中7) テキストや画像などの様々な形式の値。動的に変化することもある。

・サーバマシン(図中8) ウェブ・サーバを実行し、NLO部品やHTMLドキュメントを保持する機能をもつコンピュータ。

・クライアントマシン(図中9) ウェブ・ブラウザを実行するコンピュータ。

・イントラネット(図中10) 社内や所内など閉じたコンピュータネットワーク。

・接続可能NLO部品検索テーブル(図中11) プロキシ-NLO部品がスタブプログラムによって通知されたデータソースを使用している接続可能NLO部品の識別子を検索するためのテーブル。

【0012】ここで図2に示すフローチャートに従って本実施例の動作について説明する。

(ステップ201) ユーザがウェブ・ブラウザに表示したいHTMLドキュメントを含むURLをウェブ・ブラウザに指定する。

(ステップ202) ウェブ・サーバは指定されたURLから接続するウェブ・サーバのアドレスを取り出す。

(ステップ203) ウェブ・サーバはアドレスに指定されたウェブサーバに接続する。

(ステップ204) ウェブ・サーバはURLに指定されたHTMLドキュメントをウェブ・ブラウザに送信する。

(ステップ205) ウェブ・ブラウザは送信されたHTMLドキュメントを受信する。

(ステップ206) ウェブ・ブラウザは受信したHTMLドキュメントの中に接続可能NLO部品が含まれているかを調べる。

(ステップ207) ステップ206において接続可能NLO部品が含まれていれば、ウェブ・サーバからNLO部品をダウンロードする。

(ステップ208) ウェブ・ブラウザは接続可能NLO部品を実行する。ダウンロードされた接続可能NLO部品は図3に示すフローチャートに従って実行される。

(ステップ3001) 自分自身を初期化する。

(ステップ3002) クライアントマシンでプロキシ-NLO部品が実行中であるかを検査する。

(ステップ3003) ステップ3002においてプロキシ-NLO部品が実行されていないならば、ウェブ・サーバからプロキシ-NLO部品をダウンロードして起動する。

(ステップ3004) 接続可能NLO部品はプロキシ-NLO部品と接続する。

(ステップ3005) 接続された接続可能NLO部品は使用するデータソースの識別子をプロキシ-NLO部品の接続可能NLO部品検索テーブルに登録する。本実施例ではデータソースの識別子をD1、D2といったシンボルとしているが、それを使用するプロキシ-NLO部品が一意に識別できるものであれば何でもかまわない。

プロキシ-NLO部品はこの識別子を図4に示すような接続可能NLO部品検索テーブルに記録する。ここでN1、N2といったシンボルは接続部品NLO部品の識別子であり、ここではNLO部品" N1 " はデータソースD1~D4を使い、NLO部品" N2 " はD1~D4、NLO部品" N3 " はD1、D4、D5を使っていることを示している。

(ステップ3006) サーバマシンに対してスタブプログラムが実行中であるか検査する。

(ステップ3007) ステップ3006においてスタブプログラムが実行されていないならば起動する。

(ステップ3008) プロキシ-NLO部品はスタブプログラムと接続を行う。

(ステップ3009) スタブプログラムはローカルメモリ内に送信されたデータソースの識別子を格納する領域を作成する。

(ステップ3010) 接続が行われるとプロキシ-NLO部品は接続可能NLO部品検索テーブルの登録されている全てのデータソースの識別子をスタブプログラムへ送信する。

(ステップ3011) スタブプログラムはステップ3

7

011で送信された識別子を受信して、ステップ3009で作成された領域に格納する。

【0013】次にデータソースが更新された場合の動作について図5に示すフローチャートに従って説明する。またデータソースD2が更新された場合の動作を示した図6も参照しながら説明する。なお図6では図4の接続可能NLO部品検索テーブルを使っていた場合の動作を示している。

(ステップ5001) スタブプログラムはステップ3011で格納されたデータソースの全てについて値を読み込む。図5のようにスタブプログラムは周期的にデータソースから値を獲得する。ここでは図6のようにデータソースにD2に関して値1234がローカルメモリに読み込まれている。

(ステップ5002) 読み込んだ値が更新されているかを検査する。値が更新されていなければステップ5001にもどる。

(ステップ5003) ステップ5002で値が更新されていればプロキシNLO部品に通知を行う。図5ではデータソースD2に関してプロキシNLO部品に通知が行われている。

(ステップ5004) 通知をうけたプロキシNLO部品はスタブプログラムからデータソースの値を読み込みNLO部品内のメモリに記録する。図5ではプロキシNLO部品のローカルメモリの中に図5のようにD2の値1234が記録されている。

(ステップ5005) 接続可能NLO部品検索テーブルから更新されたデータソースを使用している接続可能NLO部品を検索する。図5のようにデータソースD2に関して接続可能NLO部品検索テーブルを検索してN1、N2の2つの接続可能NLO部品を検索する。

(ステップ5006) ステップ5005で検索された接続可能NLO部品に通知を行う。図5で検索されたN1、N2に関して通知を行う。

(ステップ5007) 更新通知をうけた接続可能NLO部品はプロキシNLO部品内のメモリから更新されたデータソースを読み込む。

以上のようにこの実施の形態のウェブ・ブラウザを介したデータ提示システムはあるデータソースが更新されたときに、そのデータソースを使用している全ての接続可能NLO部品を検索し、通知を行うことができるので、ウェブ・ブラウザを介するデータ通信量を削減することができる。

【0014】実施の形態2. また、本実施の形態によれば、サーバ・ブラウザ間のデータ通信量が主としてデータソース数のみに依存し、接続可能NLO数に対する依存性が少なくなるので、必要な通信量の見積もりが著しく容易となり、リアルタイムシステムを構築しやすくなる。実施の形態1ではスタブサーバから通知を受けたプロキシNLO部品は更新されたデータソースを使って

8

いる全ての接続可能NLO部品へ無条件に通知を行っていたが、接続可能NLO部品がプロキシNLO部品へ必要なデータソースの識別子を接続可能NLO検索テーブルに登録するときに、接続可能NLO部品が要求する更新周期も指定しておくことにより、プロキシNLO部品と接続可能NLO部品間の通知回数を少なくすることができ、実施の形態1と同様の効果が得られる。プロキシNLO部品は接続可能NLO部品が使用するデータソースに登録するときに、図7のような接続可能NLO部品検索テーブルを作成する。このテーブルには接続可能NLO部品が登録するデータソースの更新周期 $T_{Rate}$ と、プロキシNLO部品がそのデータソースに関して接続可能NLO部品に関して最後に通知を行った時刻最終通知時刻 $T_{Last}$ を格納する。ただし $T_{Last}$ の初期値は0にする。この例では接続可能NLO部品N2がデータソースD3を300ミリ秒間隔で更新しており、最後にN2に対してD3に関する通知が行われたのは時刻が78254ミリ秒であることを示している。ここで扱われる時刻はシステムが起動されてからの相対時刻である。

【0015】現在時刻 $t$ であるときデータソース $D_i$ が更新された場合、プロキシNLO部品は接続可能NLO検索テーブルから $D_i$ に関する最終通知時刻 $T_{Last}$ と更新周期 $T_{Rate}$ を得る。もし $t - T_{Rate} > T_{Rate}$ であればデータソースを要求している接続可能NLO部品へ通知を行い $T_{Rate}$ に現在時刻 $t$ を代入する。もし $t - T_{Rate} < T_{Rate}$ であれば通知は行わない。例えば現在時刻は79000ミリ秒であるとき、データソースD1が更新された場合、D1を使用しているNLO部品はN1、N2、N3、N4であるが、最終通知時刻から現在の経過時間をそれぞれについて算出するとN1は500ミリ秒経過し、N2は800ミリ秒、N3は200ミリ秒、N4は200ミリ秒が経過していることになる。N1とN2は更新周期がそれぞれ300ミリ秒であるので経過時間は更新時間を経過しているため通知を行うが、N3、N4に関しては更新周期が300ミリ秒なのでまだ更新時間に達していないので通知は行われない。以上のようにこの実施の形態のウェブ・ブラウザを介したデータ提示システムではデータソースが更新されたときに通知する接続可能NLO部品に関して、現在時刻がそのデータソースに関する更新作業が完了した時刻から経過した時間が、データソースに指定された更新時間に満たない場合は通知を行わないようにすることができる。これにより不要なデータ通信が回避され、通信量が削減できる。

【0016】実施の形態3

実施の形態1ではプロキシNLO部品がデータソースを要求している全てのNLO部品にイベントを送信していたが、プロキシがNLO部品へ通知するイベントに次に通知しなければならないNLO部品の識別子を含めることにより、同様の効果が得られる。ここでは図8

9

に示すフローチャート および図9 を使って動作を説明する。

( ステップ8001 ) ダウンロードされた接続可能NLO部品  $N_{Target}$  は、プロキシNLO部品へ使用するデータソースの識別子  $D_{Target}$  を登録する。例えば  $N_{Target}$  を  $N_2$  とすれば、 $N_2$  は  $D_1$ 、 $D_2$ 、 $D_3$ 、 $D_4$  の4つの識別子を登録する。

( ステップ8002 ) プロキシNLO部品は  $N_{Target}$  に対して、登録する全てのデータソース  $D_{Target}$  を直前に登録した別の接続可能NLO部品の識別子  $N_{Prev}$  を返す。これは  $N_{Target}$  が登録する全てのデータソースについて行われる。例えば  $N_1$  については既に登録してあるとして  $N_{Target}$  を  $N_2$  とすれば、データソース  $D_1$ 、 $D_2$ 、 $D_3$  についてはすべて  $N_1$  を返し、 $D_4$  は直前に登録したNLO部品がないので  $N_2$  自身の識別子が  $N_{Prev}$  になる。

( ステップ8003 )  $N_{Target}$  は返された識別子をデータソースごとにローカルメモリに記録する。このときデータソースの識別子と対応づけて  $N_{Prev}$  を記録する。例えばこの例についてすべてのNLO部品について登録が完了した場合、それぞれのNLO部品のもつローカルメモリ内の  $N_{Prev}$  は図9 のようになる。

( ステップ8004 ) プロキシNLO部品はデータソースそれぞれについて、最後に登録をしたNLO部品の識別子を検索する接続可能NLO部品検索テーブルを作成する。この例の接続可能NLO部品検索テーブルは図10 のようになる。

【0017】次に本実施例でデータソースが更新された場合の動作について図11のフローチャートを使って説明する。

( ステップ1101 ) プロキシNLO部品は接続可能NLO部品検索テーブルの中から最後に登録を行った接続可能NLO部品  $N_{Last}$  を検索する。

( ステップ1102 ) プロキシNLO部品は  $N_{Last}$  へ通知を行う。

( ステップ1103 ) 通知が行われた  $N_{Last}$  はプロキシNLO部品からデータソースの値をローカルメモリに読み込む。

( ステップ1104 )  $N_{Prev}$  と  $N_{Last}$  が同じであるかを検査する。

( ステップ1105 ) ステップ1105で  $N_{Prev}$  と  $N_{Last}$  が同じでなければ  $N_{Prev}$  に通知を行う。

( ステップ1106 ) 通知が行われた  $N_{Prev}$  は自分自身の識別子  $N_{Prev}$  を  $N_{Last}$  としてステップ1103に従って更新されたデータソースの値を読み込む。例えば、データソース  $D_1$  が更新された場合は  $N_4 \rightarrow N_3 \rightarrow N_2 \rightarrow N_1$  の順に通知が行われる。本実施例を使えば、プロキシNLO部品は1つのデータソースを使っている接続可能NLO部品の数によらずに高速な処理を行えるため、以降の通知作業が遅れることはない。

10

【0018】請求項3記載のウェブ・ブラウザを介したデータ提示システムでは、データソースが更新されたときに接続可能NLO部品に通知する際に、更新されたデータソースに関して通知要求している全ての接続可能NLO部品に通知することなく、1つの接続可能NLO部品だけに通知することによりデータソースを使っている全ての接続可能NLO部品がデータソースに関する値を更新することができ、ウェブ・ブラウザを介してのデータ通信量を削減できる。

【0019】

【発明の効果】本発明の第1から第3の構成によるウェブ・ブラウザを介したデータ提示システムにおいては、サーバマシン上に、データソースを一括して管理するスタブプログラムを設け、クライアントマシン上に、スタブプログラムからデータソースの値を読み込み、その値を必要としているプログラム部品を検索して送信できるプロキシプログラムを設けたので、クライアントマシン上のプログラム部品は個別にサーバマシンにアクセスすることなく、プロキシプログラムのローカルメモリからデータを読み取ることができ、ウェブ・ブラウザを介してのデータ通信量を削減することができるという効果を奏する。

【図面の簡単な説明】

【図1】本発明によるウェブ・ブラウザを介したデータ提示システムの構成図である。

【図2】実施の形態1に基づく初期時の動作を示すフローチャートである。

【図3】実施の形態1に基づく接続可能NLO部品実行時の動作を示すフローチャートである。

【図4】実施の形態1に基づく接続可能NLO部品検索テーブルである。

【図5】実施の形態1に基づくデータソース更新の場合の動作例を示す図である。

【図6】実施の形態1に基づくデータ更新における動作例を示す図である。

【図7】実施の形態2に基づく接続可能NLO部品通知テーブルである。

【図8】実施の形態3の動作を示すフローチャートである。

【図9】実施例3の  $N_{Prev}$  の登録状況を示す図である。

【図10】実施の形態3に基づく接続可能NLO部品検索テーブルである。

【図11】実施の形態3のデータ通知の動作を示すフローチャートである。

【図12】従来のウェブ・サーバを介したデータ提示システムの構成図である。

【図13】従来のウェブ・ブラウザを介したデータ提示システムの動作を示すフローチャートである。

【図14】従来例に基づくデータソース更新時の動作

11

を示すフローチャートである。

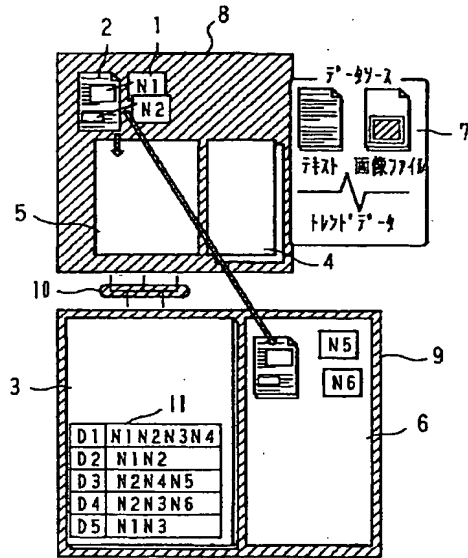
## 【 符号の説明 】

1 接続可能NLO部品、 2 HTMLドキュメント、 3 プロキシNLO部品、 4 スタンププログラ

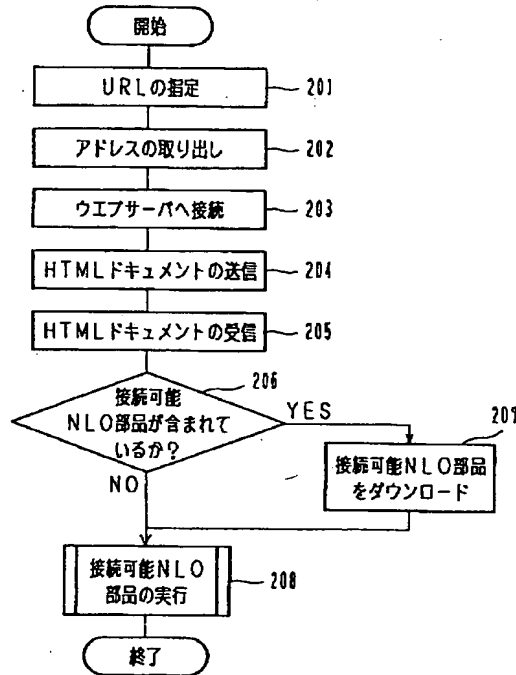
12

ム、 5 ウェブ・サーバ、 6 ウェブ・ブラウザ、 7 データソース、 8 サーバマシン、 9 クライアントマシン、 10 イン트라ネット、 11 接続可能NLO部品検索テーブル。

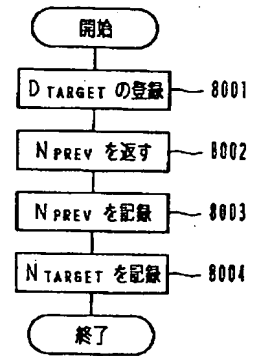
【 図1 】



【 図2 】



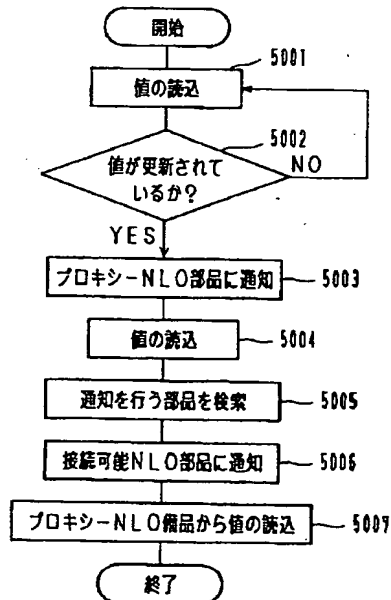
【 図8 】



【 図4 】

	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>
D <sub>1</sub>	True	True	True	True		
D <sub>2</sub>	True	True				
D <sub>3</sub>	True	True		True		
D <sub>4</sub>		True	True			
D <sub>5</sub>			True		True	
D <sub>6</sub>						True

【 図5 】

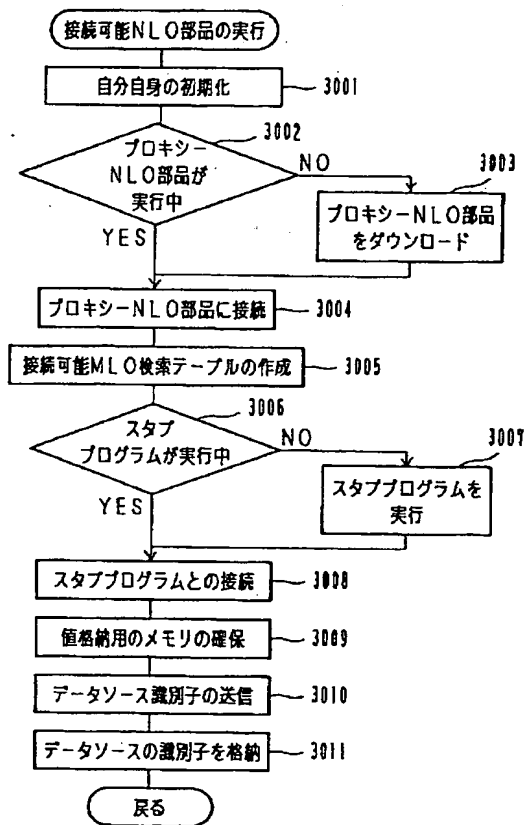


【 図9 】

N1のNPrev (D<sub>1</sub>.N<sub>1</sub>). (D<sub>2</sub>.N<sub>1</sub>). (D<sub>3</sub>.N<sub>1</sub>)  
 N2のNPrev (D<sub>1</sub>.N<sub>1</sub>). (D<sub>2</sub>.N<sub>1</sub>). (D<sub>3</sub>.N<sub>1</sub>). (D<sub>4</sub>.N<sub>2</sub>)  
 N3のNPrev (D<sub>1</sub>.N<sub>2</sub>). (D<sub>4</sub>.N<sub>2</sub>). (D<sub>5</sub>.N<sub>3</sub>)  
 N4のNPrev (D<sub>1</sub>.N<sub>3</sub>). (D<sub>3</sub>.N<sub>2</sub>)  
 N5のNPrev (D<sub>5</sub>.N<sub>5</sub>)  
 N6のNPrev (D<sub>6</sub>.N<sub>6</sub>)



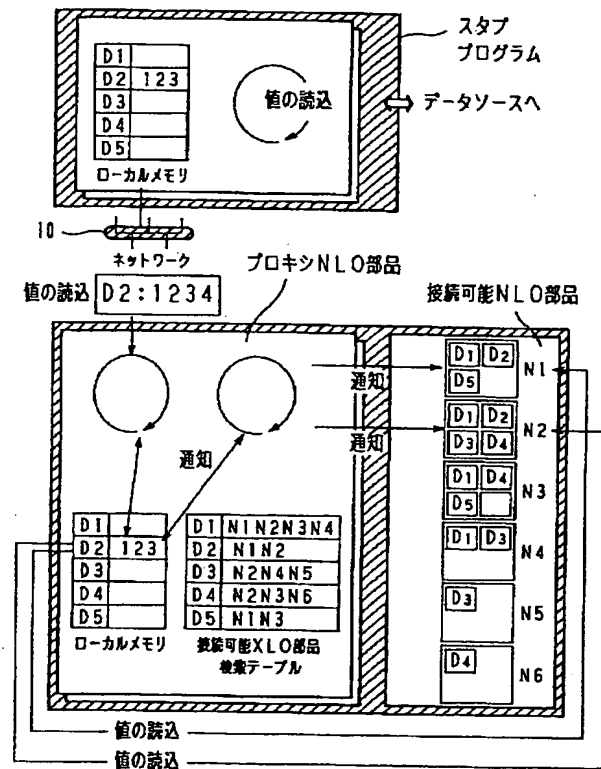
【 図3 】



【 図7 】

		N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>
D <sub>1</sub>	T <sub>Last</sub>	78500	78200	78800	78800		
	T <sub>Rate</sub>	300	300	400	400		
D <sub>2</sub>	T <sub>Last</sub>	78200	78400				
	T <sub>Rate</sub>	700	500				
D <sub>3</sub>	T <sub>Last</sub>	78800	78254		78800		
	T <sub>Rate</sub>	300	300		300		
D <sub>4</sub>	T <sub>Last</sub>		78800	78700			
	T <sub>Rate</sub>		300	400			
D <sub>5</sub>	T <sub>Last</sub>			78700		78303	
	T <sub>Rate</sub>			400		400	
D <sub>6</sub>	T <sub>Last</sub>						78403
	T <sub>Rate</sub>						400

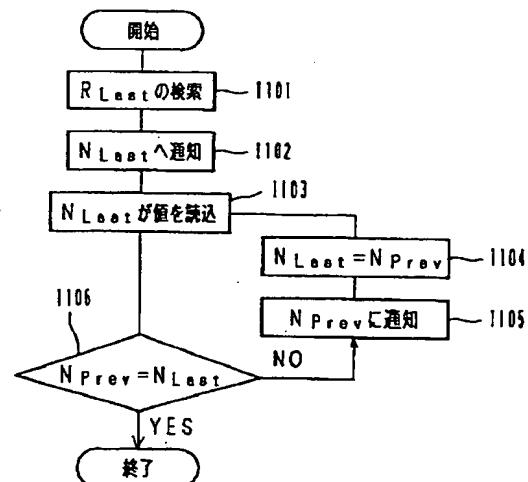
【 図6 】



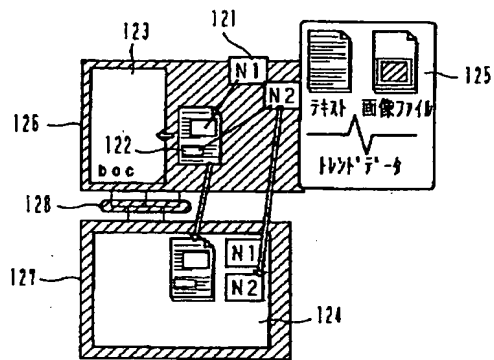
【 図10 】

データソース識別子	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>
接続NLO部品の識別子	N <sub>1</sub>	N <sub>2</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>

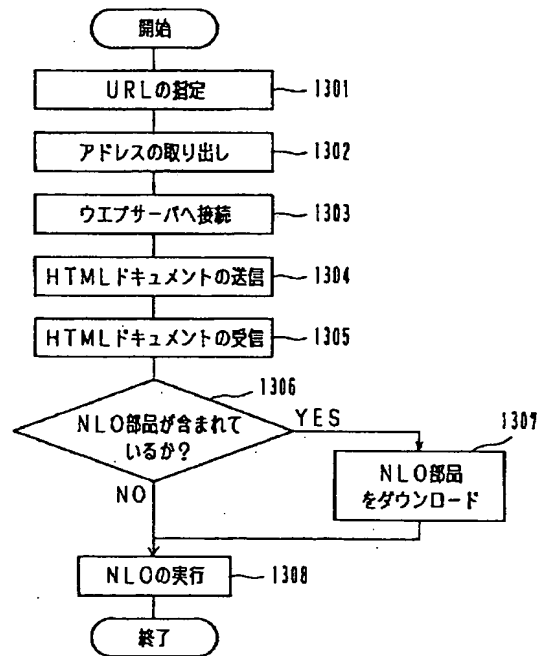
【 図11 】



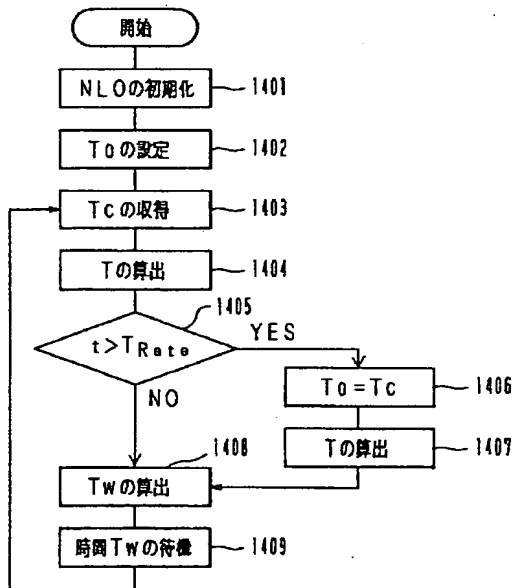
【 図12 】



【 図13 】



【 図14 】



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**